

# **A METHOD AND APPARATUS FOR TRANSMITTING DATA OVER A NETWORK**

## **BRIEF DESCRIPTION OF THE INVENTION**

### **5                   FIELD OF THE INVENTION**

The present invention relates to network data transmission systems and more particularly to methods of file transmission over a network.

### **CROSS REFERENCE TO RELATED APPLICATION**

- 10           Reference is made and priority claimed to U.S. Provisional Application entitled METHOD AND APPARATUS FOR TRANSMITTING DATA OVER A NETWORK, filed September 14, 2001, bearing application number 60/322,595. Also this patent application is related to U.S. patent application number 09/709,948 entitled METHODS FOR PROVIDING VIDEO-ON-DEMAND SERVICES FOR BROADCASTING
- 15   SYSTEMS, filed November 10, 2000, U.S. patent application number 09/584,832 entitled NON CLIENT SPECIFIC ON-DEMAND DATA BROADCAST (As Amended), filed May 31, 2000. Each of the above referenced applications are hereby incorporated by reference herein.

### **20                   BACKGROUND OF THE INVENTION**

The Internet has expanded greatly since the beginning of the 1990's and Internet traffic continues to grow exponentially as more users and newer uses are being added to this worldwide system everyday. Simple text file transfers have been replaced by sophisticated web pages including music, graphics, animation and video clips.

- 25           Increasing traffic on a network such as the Internet, along with larger file sizes has resulted in increasingly greater traffic congestion and longer access and download times.

One solution for the reducing the network congestion problems has been a brute force approach. Faster communications devices and better transmission mediums such as coaxial cable and optical lines have been used to increase the bandwidth and efficiency of the network systems. However, replacing existing cables and networking equipment with higher bandwidth and more efficient systems is costly. Furthermore, the rate of increase in network traffic is surpassing the rate at which technological improvements are being introduced.

Another limitation of the standard network communication model is that each client requires its own bandwidth. Thus, the total required bandwidth is directly proportional to the number of users. Cache memory within some local servers such as proxy servers have been used to improve bandwidth limitations but using cache memory does not solve the problem because cache memory is also limited.

Thus it is desirable to provide a system that is capable of improving network efficiency and freeing up available bandwidth, without expensive hardware improvements.

## SUMMARY OF THE INVENTION

Briefly, one aspect of the present invention is embodied in a method of operation for a system for transmitting and receiving data within a network comprising dividing a file into its dynamic data components and static data components by comparing at least one expiration parameter corresponding to each component to a set setting a threshold, transmitting a request for a file from an intelligent client, receiving the file request by an intelligent server, transmitting at least the dynamic data components of the requested file to the intelligent client, and recreating by the intelligent client, the requested file by recombining the static components and the dynamic components of the requested file.

Another aspect of the present invention is embodied in a method of parsing a data file into its static and dynamic components, comprising setting a threshold value for separating static and dynamic data components, obtaining expiration parameters for each portion of a data file, comparing each expiration parameter to the threshold value, identifying as dynamic data the portions of the data file that have an expiration value

lower than the threshold value, and identifying as static data the portions of the data file that have an expiration value higher than the threshold.

Another aspect of the present invention is embodied in a method of sending requested data to a client from an intelligent file server, comprising receiving a request for a file from a client, the file including static data components and dynamic data components, determining whether a client is intelligent, creating static data packets containing static data components and separate dynamic data packets containing dynamic data components, and transmitting static data components and dynamic data to the intelligent client.

Another aspect of the present invention is embodied in a method for processing data received from an intelligent server comprising, sending a request for a file from an intelligent client, over a network, to an intelligent server, receiving static data packets and dynamic data packets separately, reconstructing the requested file from its static data packets and its dynamic data packets.

Another aspect of the present invention is embodied in a method of creating an intelligent protocol data packet comprising, including identifying information identifying the intelligent data protocol.

Another aspect of the present invention is embodied in a method for transmitting data over a network comprising, requesting a file from a server connected to a network by a client connected to a network via a network connection provider system, capturing the file request by the server providing the network connection, checking a memory cache related to the network connection provider for the presence of a first saved data relating to the requested file, sending static data saved on the cache memory, sending a request to the server for all data not saved on the cache memory, transmitting all data not saved on the cache memory to the network connection provider, transmitting all data not saved on the cache memory to the client, and reconstructing the requested file from static data and dynamic data.

## BRIEF DESCRIPTION OF THE DRAWINGS

Prior art FIG. 1 illustrates a traditional web Internet system;

Prior art FIG. 2 is a flow diagram for the processes of transmitting a file over the Internet;

Prior art FIG.3 is a representation of a packet under a TCP/IP protocol;

5 Prior Art FIG. 4 is a representation of web page level interactions between a client and a server;

FIG. 5 is an illustration of a data transmission system according to one embodiment of the present invention;

10 FIG. 6 is a flow diagram of the operation of one embodiment of the new data transmission system;

FIG. 7 shows the new technology server of the system of FIG.1 in more detail according to one embodiment of the present invention;

FIG. 8 is a flow chart for the operation of one embodiment of an intelligent data server;

15 FIG. 9 is an illustration of one embodiment of an intelligent ISP server;

FIG. 10 is a flow diagram of the operation of one embodiment of an intelligent ISP server;

FIG. 11 is an illustration of one embodiment of an intelligent client;

20 FIG. 12 is a flow diagram of the operation of one embodiment of the intelligent client;

FIG. 13 illustrates the static and dynamic data content of a typical file;

FIG. 14 is a flow diagram for the process of dividing a data file into its dynamic and static components for one embodiment of the present invention;

25 FIG. 15 illustrates a data packet according to one embodiment of the present invention;

FIG. 16 is a flow diagram of the operation of a system based on one embodiment of the present invention.

## DETAILED DESCRIPTION OF THE INVENTION

5

Prior art FIG. 1 illustrates a traditional Internet web system. Referring to FIG.1, a series of users "user1" 12 to "user N" 14 are connecting to the Internet through Internet Service Providers (ISP). A typical private user such as "user1" may connect to the Internet through the IPS servers 16, or 20. Business users may connect to the Internet through their intranet network and may pass through firewalls and/or proxy servers. Alternatively, users belonging to proprietary ISPs such as America Online or Prodigy may direct requests from their subscribers through a gateway and possibly a proxy server connecting their intranet to the Internet. In each case a user makes a request 13 for a web page. The request 13 travels through various intermediate routers 24 and 28, and various links (e.g. satellite links) such as shown by 32 and 36 before reaching its final destination. Normally, the final destination is the web server 40 hosting the requested web page, where the web server responds to the user's request by sending the requested web page and/or the corresponding information to the user.

The Internet is primarily based on a packet switching technique. In packet switching technique, a large file is parsed into smaller size packets. Each packet includes a header containing information such as the destination of the packet, its source and an identification number. Each packet may be transmitted via a different route, independent of the others. Once the all the packets arrive at the destination, they are reassembled based on the ordering information within the header.

The response 42 of the web server 40 will be routed via different routes based on the level of congestion in the system and the value of parameters of various routes. Newer routers using the Open Shortest Path First (OSPF) protocol are able to check a variety of parameters such as the number of hops, delay, capacity, throughput, and reliability of the circuit connecting the routers, in order to select the route by which to

send the data. The response from the web server 40 may have to be channeled through the Internet via multiple routers 26 and 30, and possible links 34 and 38 to the ISP servers 18 and 22 for each user 12 and 14.

In some cases, when private networks are connected to the Internet through a proxy server, the proxy server 18 or 22 may at least temporarily cache the most requested web pages in order to avoid using the Internet to get these web pages from the web server 40.

Prior art FIG. 2 is flow diagram for the processes of transmitting a file over the Internet. At block 44, a web server receives a file request from a client. At block 46, the web server retrieves the file from its cache memory, breaks the file into appropriately sized packets according to a TCP/IP protocol, and transmits the individual packets over the Internet. The term Cache memory as used here refers to temporary storage on a hard disk, RAM memory or any other type of memory that may be used to store data on a temporarily. At block 48, a client receives all the packets transmitted to it over the Internet, and does an error check to see if all the packets have been received. At block 49, if all the packets for a given file have not been received, the client sends a second request for the particular packets. At block 50, the client sends an acknowledgment for each packet that it has received. At block 52, after all the packets have been received, the packets are reassembled to construct the requested file.

Prior art FIG.3 is a representation of a packet under a TCP/IP protocol. Data 58 transmitted over the Internet is encapsulated according to the IP protocol prior to transmission. Headers and trailers are added to each data packet giving information about the content of the packet. The encapsulation of the data 58 in a header and trailers forms a frame and is similar to placing a letter in an envelop before mailing. However, in the case of a frame, the packet 58 is encapsulated with information necessary for routing the packet such as the origination address and destination address. Furthermore, additional data is included in the header or tail concerning various parameters such as the packet length, checksum for verifying errors occurring during transmission, and the order of the packets necessary for reassembly once the packets arrive at their destination.

Referring back to FIG. 3, a header 54 is added to the data packet providing the information necessary based on the network layer Internet Protocol (IP) layer. Similarly, the Transport layer protocol TCP requires the addition of a header and/or tail to the packet prior to transmission. The IP layer only handles the routing of the packets 58 but it does not guarantee any the delivery of the packet and simply discards any data packets if problems occur. The Transmission Control Protocol (TCP) provides a considerable number of services to the IP layer as well as the layers above the TCP layer. Most importantly, the TCP layer provides a connection-oriented protocol that enables an application to know that a packet sent over the Internet is received in its entirety at the destination. TCP receives messages from the destination acknowledging the receipt of a complete file and ensuring that the transmission was complete. Furthermore, the TCP communicates with the applications in the upper layers as well as the network systems below it such as the IP layer. In prior art FIG. 3, additional header and tail data 56 are added by the TCP layer to the packet, before transmission.

Prior Art FIG. 4 is a representation of web page level interactions between a client and server. The client computer 60 may be running a web browser 62 such as the Microsoft® Internet Explorer ® or Netscape® Navigator®. Typically, a web browser 62 assembles and displays the files sent from the server 66 to the client 60 upon the request issued by the client computer 60. The client computer 60 connects to the desired server 66 using a Universal Resource Locator (URL) that acts as an address useful in locating the server 66. One of the most popular protocols used is the Hyper Text Transfer Protocol (HTTP) used on the World Wide Web (www). Other URL protocols such as the File Transfer Protocol (FTP) may be used to retrieve files over a network, such as the Internet. In the example of FIG. 4, using an HTTP URL, the client computer 60 sends a request to the server 66 for a Hyper Text Markup Language (HTML) file, namely a HTML web page 64. Once the web browser 62 has received all the packets for the file comprising the HTML web page 64, it assembles the web page and displays it on the client's computer monitor. Additional plug-in programs 65 may be operating with the web browser 62 to perform operations a basic web browser may not be able to perform.

The server 66 may be using a web server such as the Microsoft™ IIS web server 68 as its web server running on a Windows NT Server 68 operating system. The HTML file 64 the client has requested may include static HTML pages 72 as well as special programs such as animated graphics 74 or movies on the user's computer that would be downloaded to the client and run there using special plug-in programs 65.

FIG. 5 is an illustration of a data transmission system according to one embodiment of the present invention. Client A 78 and client B 80 send requests for data through the Internet Service Provider server 82 and the network 84 to the server 86 that has the requested file 88. Client N 90 may be part of a private network 92. Client N 90 may use the private network 92 and the public network 84 to send its file requests to the server 86. The private network 92 may be connected to the public network 84 and protected from it by a firewall and/or proxy server 94.

Client A 78 may include a new technology browser that allows client A 78 to receive and process packets are formed based on the new protocol. Client B may have an old technology browser and would have to receive packets based on the old protocol in order to process and reassemble the packets it receives. The ISP server 82 may also be a new technology server capable of receiving, processing and transmitting packets based on the new protocol. Additionally, an intelligent ISP server 82 may include cache memory 83 that might be used to store the static data portions of the most requested data files. As used herein, the term "intelligent" machine, such as an intelligent server, or an intelligent ISP or an intelligent client refers to a machine that is capable of processing a data file that may be divided up into its time sensitive (dynamic) and time insensitive (static) data parts. Processing may include parsing a data file into its components, transmitting, receiving, and reassembling the data file. In contrast, the term "non-intelligent" as used herein in this document refers to a client, file server or intermediate node server such as an ISP sever that is based on the current state of the art technology and does not parse data into its static and dynamic components.

The public network 84 may still operate based on the existing protocol. In one embodiment, the public network 84 may be the Internet and the governing protocol may be the TCP/IP protocol. However, it should be noted that the teachings of the present invention are not limited to a particular protocol and/or a particular network type.



The server 86 is a new technology server capable of classifying, separating and processing data files into their dynamic and static components. A file 88 residing on the server 86 may be divided into dynamic data and static data components by the server software.

The distinguishing characteristic between dynamic data and static data is based on the definition that dynamic data is client specific, and static data is considered to be non-client specific data. Static data content changes at such a slow rate that the server can treat it as static or constant. In contrast, dynamic data content changes so rapidly that the server must handle it as variable data.

It should be noted that the definition of static and dynamic data is relative. A threshold may be set to separate dynamic data from static data. That is data with a rate of change greater than a set threshold may be treated as dynamic, whereas data with a rate of change below a certain rate may be treated as static data.

In the example of a web page file, the web designer may include in its design expiration dates for data comprising various parts of the web page. Things such as banners, logos, background and some text may remain have an expiration date set from days to months. On the other hand, if the web page includes a table of real time market stock values, the constantly changing stock values may have expiration dates of several seconds. In this example, if the web server threshold is set at one minute, any data within the web page file that has an expiration date of greater than one minute may be classified as static data. At the same time, any data such as the stock values with expiration dates less than one minute will be treated as dynamic value.

In one embodiment, the threshold value of the server 86 may be adjusted based the level of service requested from the server. The lower the threshold value, the greater the amount of static data and the less work there would be for the server.

Client N 90 and the ISP server 94 may also have new technology software and be capable of receiving and processing packet packets based on the new protocol.

FIG. 6 is a flow diagram of the operation of one embodiment of the new data transmission system. At block 96, a file request is transmitted by new technology client such as client A 78 (see FIG.5) to the new technology server 86. At block 98, the file request is routed through an ISP sever. If the ISP server is an intelligent server such as

ISP server 82 (see FIG. 5), the ISP server may process the request initially. At block 100, the intelligent ISP server 82 checks its cache memory 83 to see if it is storing the static data relating to the requested file. If the cache memory 83 does have the static data for the particular file requested, the ISP server might transmit the static data to the client 78, at block 102. If the static data for the requested file is not available at the cache memory 83, the ISP server transmits a request for the static and dynamic data for that file, at block 104. If the static data was stored at the cache memory 83, a request for the dynamic portion of the data is transmitted to the server 86.

If the static data is available on the cache memory, the ISP server might directly send the requested static data to the requesting client. However, this transmission method requires increasing bandwidth as the number of clients placing a request for the given data increases.

In one alternative embodiment, the static data available is broadcast over the network to the multitude of requesting clients. In an alternative embodiment, the available static data is multicast to the requesting clients. In an exemplary embodiment, at the file server 86 side, sending data to a client to provide data-on-demand services using a method of multicasting can be achieved through the steps comprising of: receiving a data file, specifying a time interval, parsing the data file into a plurality of data blocks based on the time interval such that each data block is displayable during the time interval, determining a required number of time slots to send the data file, allocating to each time slot at least a first of the plurality of data blocks and optionally one or more additional data blocks, such that the plurality of data blocks is available in sequential order to a client accessing the data file during any time slot, and sending the plurality of data blocks based on the allocating step.

In one embodiment, the parsing step includes the steps of: determining an estimated data block size, determining a cluster size of a memory in a channel server, and parsing the data file based on the estimated data block size and the cluster size. In another embodiment, the determining step includes the step of assessing resource allocation and bandwidth availability.

At block 106, intelligent server 86 receives a request for data. As discussed above, the request may be for both static and dynamic data or only for the dynamic

portion of the data. At block 108, the server 86 transmits the static and/or dynamic data based on the request it received. At block 110, once the ISP server 82 receives the transmitted data from the intelligent server 86, it forwards the data to the requesting client 78. At block 112, the requesting client assembles the static and dynamic data to recreate the file it has requested.

FIG. 7 is a new technology server according to one embodiment of the present invention. In one embodiment of the server 86, a server software 114 compliant with the existing protocol may comprise the basic operating software of the server. However, the add-on server software 116 may be providing the additional capabilities necessary to classify static and dynamic data, parse a file into its static and dynamic components, packetize packets of appropriate size based on existing network protocols, while including header or trailer information allowing the identification of the new packets comprising the static and dynamic data. The format of a packet for use in the invention is described further in FIG. 10.

FIG. 8 is a flow chart for the operation of one embodiment of an intelligent data server. At block 120, the intelligent server 86 may set a threshold value based upon which the dynamic and static data classification may be made. In one embodiment, the value of threshold may be adjusted based on the performance requirements of the web server. A lower threshold value may be used to reduce the operating burden of the intelligent server 86 by classifying a larger portion of each file as comprising of static data and requiring less frequent retransmission. At block 122, the intelligent server 86 parses the files residing in its file storage database into their dynamic and static data components based on the set threshold value. At block 124, the intelligent file server 86 receives a request from a client machine. The term intelligent file server as used herein refers to a file server capable of parsing a data file into its static and dynamic components, by setting a threshold value, and comparing each data file component having an expiration parameter to the threshold, and separating the file into its static and dynamic data based on whether a file component expiration parameter value is greater than the threshold value or less. Once the intelligent server 86 has received a request, it may ascertain whether the request came from an intelligent client at block 126. In one embodiment, at block 128, if the request came from a traditional client, and there are no

intelligent intermediate machines along any path between the intelligent server and the traditional client, then the server 86 recombines the dynamic and static portions of the requested file transmits the entire requested file based on the old network protocol. In one embodiment, the file server 86 may be a web server, the network may be the Internet and the applicable protocol may be the TCP/IP protocol.

At block 132, the intelligent web server determines whether there are intelligent intermediate machines such as intelligent ISPs along a particular route between the intelligent web server and the intelligent client. In one embodiment of the present invention, the intelligent server 86 may preferably use the route that includes at least one other intelligent device when sending its packets. In one embodiment, the determination of the presence of intelligent devices along a possible link of the route to the client is determined by using an address resolution table that includes additional fields indicating the whether an intelligent device may be present at that address. At block 134, if there are no intelligent intermediate machine along the route from the server 86 to the client 78, the server 86 may select the route based on such parameters as the congestion of the route, the number of hops, and so on.

A block 136, the intelligent web server may route the response to a first request form a client via a route including at least one intelligent intermediate device.

At block 138, if the original requester is an intelligent device, or if there are intermediate intelligent devices along the route from the intelligent server and the original client, the intelligent server transmits static data and dynamic data separately.

FIG. 9 is an illustration of one embodiment of an intelligent ISP server. In one embodiment the intelligent ISP server 82 includes an ISP operating software 140 that may be 100 percent compatible with the existing technology. Additionally, an add-on or plug-in software component 142 may be added to the ISP server 82 to allow it to receive, process and transmit the new protocol packets based on separate dynamic and static data components. Additionally, an intelligent ISP server 82 may include cache memory 83 that might be used to store the static data portions of the most requested data files.

The cache memory 83 allows at least part of the static data transmitted by a first user to be temporarily stored at that node. This allows for multiple requests received for a particular web page or file to be at least partially fulfilled by the data present in the cache

memory. For example, if the file requested by the client includes 100% static data, a second or third client request for that file does not have to travel back to the web server on which the file resides. Since a copy of all static data from the first request for this web page is cached on at least one intelligent device along the route between the intelligent client and the intelligent web server, the intermediate device may be able to supply the requested and fulfill the response without going all the way back to the host web server. Traffic for at least a part of the network is reduced. This scenario may even be implemented in cases of requests coming from traditional clients.

- Alternatively, if the file cached at the intermediary node contains both static and dynamic data, the intermediate intelligent machine may send a request for the dynamic data only. If the initial request came from an intelligent client, the intermediate intelligent machine may send dynamic and the static portions of the data in separately. In an alternative embodiment, the intelligent intermediate machine may ascertain whether the static data portions of the requested file are cached at other intelligent intermediate machines along the route between this intelligent machine and the original intelligent client. In that case, the first intelligent machine may only transmit the dynamic portion of the data along to the second intelligent machine. In either scenario, the transferred files are smaller resulting in less traffic congestion on the Internet.

- Also, if the original client is not an intelligent client, once the intelligent intermediate machine receives the dynamic components of the file, it combines them into a single file and transmits them according to the existing TCP/IP protocol.

It should be noted that the reductions in traffic resulting from fewer requests and smaller size file transmissions would multiply with the presence of increasing numbers of intelligent machines on the network.

- Furthermore, once a first request for a file (e.g. web page) is fulfilled, if the intelligent client requests further updates of the same web page, the request would be for the dynamic data portion of that file only. Since in most cases, the ratio of the dynamic data portion of a typical web page as an example is a lot smaller than the static portion of the data, the amount of the data traveling over the network is reduced. The number of requests and the amount of data requested from the web server may also reduce because

intermediary intelligent devices may contain most or all of the data requested and respond to the request.

FIG. 10 is a flow diagram of the operation of one embodiment of an intelligent ISP server. At block 146, the intelligent ISP server 82 receives a file request from a client. At block 148, the ISP server 82 determines if the claim came from an intelligent client 78 or an old technology client 80. At block 150, if the request was initiated by an intelligent client 78, the ISP server checks its cache memory for the availability of the static data to be sent to the intelligent client. The term "Intelligent client" as used here refers to a client that is able to receive data files packets divided into static and dynamic data components and can recreate the complete file by recombining both the static and dynamic data files. At block 154, the ISP server transmits a request for dynamic data to the intelligent file server 86. If the static data is not present in the cache memory, the request transmitted at block 154 would also include a request for the static data. At block 156, the ISP server 82 receives the data that may include both static and dynamic data. At block 158, the ISP server 82 transmits the dynamic and static data to the requesting intelligent client 78.

Alternatively, at block 160, if the request came from client 80, who is not able to process the new protocol based on separate dynamic and static data, the ISP server 82 forwards the request from client 80 to the file server 86. In one embodiment, if the ISP server has the static portions of the requested file in its cache memory 83, it may only forward a request for the dynamic portion of the data relating to the requested file. At block 162, the ISP server receives the dynamic and the static data, if requested. At block 164, the ISP server 82 may recombine the dynamic and static portions of the requested file. At block 166, the ISP server transmits the recombined requested file to the non-intelligent client 80 using the existing transmission protocol.

In one embodiment, the intelligent ISP server 82 may be receiving data files from non-intelligent file servers. The term "intelligent ISP" as used here refers to an ISP server capable of processing data files packets that are divided into their static and dynamic components, by recombining a file from its static and dynamic components. Under that scenario, the ISP server 82 may still be parsing the requested file into its dynamic and static components and transmitting the file as such to its intelligent clients.

FIG. 11 is an illustration of one embodiment of an intelligent client 78. In one embodiment, the intelligent client machine 78 may include client-operating software 168 based on the existing (old) network protocol, and allowing the client to access the network using this software. In one embodiment, the client-operating software may a web browser allowing the client to access the Internet.

The intelligent client machine 78 may include an add-on or plug-in type of software 170 that enables the intelligent client machine to receive, detect, process and use the new protocol based on the dynamic and static data definitions. This implementation allows for backward compatibility of the intelligent client machine with the existing network and its operating protocol. Furthermore, the plug-in upgrade may allow the conversion of an old type client machine into an intelligent client machine by simply installing a software upgrade.

FIG. 12 is a flow diagram of the operation of one embodiment of the intelligent client. At block 172, the intelligent client 78 sends a request for a file to an intelligent server. The request may be partially satisfied at the ISP server as described above. The intelligent client receives the static and dynamic portions of the file at block 174. The client 78 recombines the static and dynamic data packets into the requested file.

In one embodiment, the static data may be broadcast to the intelligent client 78. The plug-in software 170 is capable of capturing and recombining the static data packets that are presented to it by the intelligent ISP server 82, based on a packet-multiplexing table.

FIG. 13 illustrates the static and dynamic data content of a typical file. A typical requested file 88 such as the one illustrated here may be divided up into its dynamic data components 180 and its static data components 178. As described above, the static data may be defined based on its content and the rate of change of its content over time. This static data 178 is defined as non-client specific since its content does not vary based on the time the client places a request for that data. In contrast, the dynamic data 180 may have content that may change at a much higher rate than the static data and subsequently, at each time the client request is made, the data content may have changed. This data is classified as client specific data. The ratio dynamic data to static data is dependent may vary based on the defined threshold used to divide dynamic and static data.

FIG. 14 is a flow diagram for the process of dividing a data file into its dynamic and static components for one embodiment of the present invention. At block 182, a threshold value is set for separating static data components and dynamic data components. The threshold value would be used to compare against an expiration parameter corresponding to each of part of a file. At block 184, the file server 86 may obtain the expiration parameters for each portion of the data in a particular file. At block 186, the file server 86 may compare the expiration parameter for each portion of the data file against the threshold value. At block 190, a dynamic data portion may be identified as the portion of the data file where the corresponding expiration parameter is less than the threshold value of the file server 86. Alternatively, at block 188, if the expiration parameter of that portion of the data is measured to be higher than the threshold value, then that portion of the data may be identified as static data. In one embodiment, the operation of dividing a file into its dynamic and static components may be done offline, prior to receiving a request for that particular file.

FIG. 15 illustrates a data packet according to one embodiment of the present invention. In one embodiment, an identifying header 198 is added to the data packet 192 providing the information necessary for the operations of the network layer based on Internet Protocol (IP) 196 in identifying and transmitting the packet. Alternatively, the Transport layer protocol TCP 194 may include the addition of a header and/or trailer 200 to it prior to transmission. The new data packet may be distinguished from an old type data packet by inserting identifying information 200 in the (TCP) header 194. The additional bits of data added would be added to locations of the header 194, that is not currently used in the TCP protocol. This enables backward compatibility with existing technology and hardware. The new technology routers, ISP's and clients would be able to use the information inserted 198 or 200 in the TCP header 194 or in the data content 192 in order to recognize a new technology data packet from an old type data packet.

FIG. 16 is a flow diagram of the operation of a system based on one embodiment of the present invention. At block 202, a file is divided up into its dynamic and static data components by comparing at least one expiration parameter corresponding to each data part to a threshold value. The threshold value may be set at the file server based on



an analysis of the current load of the server. In one embodiment of the present invention, threshold may be adjusted based on the changing file server traffic. A higher threshold would result in classifying a larger portion of a file as static data, which may result in a lower file server traffic.

- 5           At block 204, an intelligent client may transmit a file request to an intelligent server. At block 206, the intelligent file server receives the request from the intelligent client. At block 208, the intelligent file server transmits the static data and/or dynamic data parts of the requested file to the requesting client. In one embodiment of the present invention, the intelligent file server may transmit a copy of the requested file that has not  
10   been divided into its static and dynamic components, whenever the file server recognizes that the requesting client is not an intelligent client. In another embodiment, the intelligent file server may start broadcasting the static data components of the requested file once the number of requests for a particular file have exceeded a specified number. At block 210, the intelligent client may receive the requested file and recreate the file by  
15   recombining it from its static and dynamic components.

The foregoing examples illustrate certain exemplary embodiments of the invention from which other embodiments, variations, and modifications will be apparent to those skilled in the art. The invention should therefore not be limited to the particular embodiments discussed above, but rather is defined by the following claims.